Data Cleaning for Gender Inequality Insights

Elisabet Arelly Sulú Vela
Data Engineering
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 4448. CP 97357
Ucú, Yucatán. México
Email: 2309212@upy.edu.mx

Karen Cardiel Olea
Data Engineering
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 4448. CP 97357
Ucú, Yucatán. México
Email: 2209039@upy.edu.mx

Lester Stephan Estrada López
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 4448. CP 97357
Ucú, Yucatán. México
Email: lester.estrada@upy.edu.mx

Abstract

This report describes the data processing steps taken with the "`gender_inequality_index.csv`" dataset, which contains indicators of gender inequality in different countries. The goal of the processing was to clean and prepare the data for better analysis. We did several things, such as removing unwanted values, elimination of duplicate data, converting columns to numbers, handling missing values, and creating new important features. These changes help us analyze patterns in gender inequality more accurately and understand the factors that affect women's situations in different places.

Index Terms

Data Preprocessing, Gender Inequality Index, dataset, data visualization, data cleaning

Data Cleaning for Gender Inequality Insights

I. Introduction

Data preprocessing is an important part of preparing data for analysis. Before data can used to answer questions or solve problems, it must be clean and organized. This is obtain by fixing errors, handling missing information, and changing data types so they are correct. Good preprocessing helps to get better results and clearly understand the data.

This report has focus on preprocessing techniques used for a dataset about gender inequality. It will be explained the steps taken to clean the data, change data types, and create new features that helps to analyze the information better. By using these techniques, the aim is to prepare the dataset for a thorough analysis, to further on explore important patterns and trends in gender inequality.

II. Objective of the Case

The goal of preprocessing this file in this case is to clean and prepare the dataset for a correct analysis and plotting of the gender inequality indicators. This is achieved with a Python code that is able to perform data preprocessing of missing values, convert data types, and create new features to improve understanding. The goal is to have the data correctly filled or removed (depending on the case), which will allow to perform exploratory analysis and obtain more statistical information, such as visualizing relationships between factors like maternal mortality, adolescent birth rates, and female participation in education and the labor force.

III. Preprocessing Steps

A. Importing libraries

Before starting it is important to import the necessary libraries. In this case, `pandas` and `numpy` are needed to manipulate data, whereas `matplotlib.pyplot` and `seaborn` are needed for data visualization, to create plots, for example. Also, since this project was realised using Google Colab, `google.colab` is used to give access to Google Drive, where the dataset is stored.

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from google.colab import drive
```

B. Loading dataset

The first code line of the next extract of the program uses the function `drive.mount()`, which allows the Colab Notebook to have access to Google Drive. Next, the CSV file of the dataset is loaded and read with `pd.read_csv()`. At the end, the `encoding='latin1'` parameter is used to ensure special characters are represented correctly and to avoid encoding errors when reading the file.

```
    # Load the dataset
drive.mount("/content/drive/")
df = pd.read_csv("/content/drive/MyDrive/
    datasets/gender_inequality_index.csv",
    encoding='latin1')
```

C. Inspecting Columns

`df.head()` prints the first five rows of the data set, this provides a quick insight into it and the type of data that each column has. It is followed by `df.columns`, that shows all the columns titles from the data set. Finally, the last column is erased using the `drop()` function, specifying with the `axis=1` parameter that `Unnamed: 12` is a column, not a row.

```
df.head()
df.columns
df.drop("Unnamed: 12", axis=1, inplace = True)
```

With the following extract of the program, several columns are converted to numeric types with the `pd.to_numeric()` function, and for all values that cannot be converted to numeric within those columns, they are converted to `NaN`.

```
# Convert appropriate columns to numeric types
     (error coercion)
df['Maternal_mortality'] = pd.to_numeric(df['
    Maternal_mortality'], errors='coerce')
df['Adolescent_birth_rate'] = pd.to_numeric(df
    ['Adolescent_birth_rate'], errors='coerce'
    )
df['Seats_parliamentt (% held by women)'] = pd.
    to_numeric(df['Seats_parliamentt (% held by
    women)'], errors='coerce')
df['F_secondary_educ'] = pd.to_numeric(df['
    F_secondary_educ'], errors='coerce')
df['M_secondary_educ'] = pd.to_numeric(df['
    M_secondary_educ'], errors='coerce')
df['F_Labour_force'] = pd.to_numeric(df['
    F_Labour_force'], errors='coerce')
df['M_Labour_force'] = pd.to_numeric(df['
    M_Labour_force'], errors='coerce')
```

D. Basic Cleaning Data

All the `'..'` contained in the dataset are missing values that must be handled. They are replaced with `NaN` with the `df.replace()` function. Then, every `NaN` value is removed from the Data Frame by using the `df.dropna()` function.

```
# Basic data cleaning (replace '..' with NaN)
df.replace('..', np.nan, inplace=True)
# Drop rows with missing values
df.dropna(inplace=True)
```

E. Dropping duplicates

The function `df.drop_duplicates()` is used, as its name suggests, to eliminate any duplicated column, and by adding the `(subset=['HDI rank'])` parameter, the function is only considering the `HDI rank` column.

```
df = df.drop_duplicates(subset=['HDI rank'])
```

F. Creating a new feature

The `maternal_to_adolescent` feature is created with the ratio of `maternal_mortality` and `Adolescent_birth_rate`

```
df['maternal_to_adolescent'] = df['
    Maternal_mortality'] / df['
    Adolescent_birth_rate']
```

Then, the created ratio is categorized into bins with `pd.cut`, assigning labels to these bins.

```
bins = [0, 1, 5, 10, 20, 50, np.inf]
labels = ['0-1', '1-5', '5-10', '10-20', '
    20-50', '>50']
df['maternal_to_adolescent_binned'] = pd.cut(
    df['maternal_to_adolescent'], bins=bins,
    labels=labels)
```

IV. Algorithms and results of cleaning and transformation

To visualize and explore data, several visual elements are printed, this step is known as Exploratory Data Analysis.

Here are some functions that will be used in the next extracts of code:

- `plt.grid()`: to add a grid to a plot in matplotlib.
- `plt.title()`: to change the title of a plot.
- `plt.show()`: to print a plot.

The first visual element generated is a histogram that represents the distribution of the percentage of parliamentary seats held by women in different countries. Each bar in the graph shows the number of countries that have a certain percentage of female representation in their parliament. This is how Fig. 1 was obtained:

```
# 1. Histogram of Seats held by women in
    Parliament
sns.histplot(df['Seats_parliamentt(%_held_by_
    women)'], color = '#8C2981')
plt.grid(True, axis='y', color='gray',
    linestyle='--', linewidth=0.5)

plt.title('Histogram:_Seats_held_by_women_in_
    Parliament_(%)')
plt.show()  # Show plot
```

Fig. 2 presents the boxplot created for the percentage of Females that completed Secondary Education. This graph indicates that most countries have between 40 percent and 80 percent of women completing secondary education, with a median value around 60 percent. Some countries have percentages close to 100 percent, while others are closer to 20 percent. The boxplot was printed using the following code:

```
# 2. Boxplot for F_secondary_educ
sns.boxplot(df['F_secondary_educ'], color = '#
    FB8761')
plt.grid(True, axis='y', color='gray',
    linestyle='--', linewidth=0.5)

plt.title('Boxplot:_Female_Secondary_Education
    _(%)')
plt.show()  # Show plot
```

In Fig. 3 is presented a Count Plot of the Maternal Mortality to Adolescent Birth Rate ratio that was created before as a new feature. The image shows a bar chart representing the relationship between maternal mortality and the adolescent birth rate, classified into different ranges or groups (binned). The x-axis shows the ranges of the relationship between maternal mortality and the adolescent birth rate, while the y-axis shows the number of countries (count) that fall within each of those ranges. Here is the code that was used:

```
# 3. Create a count plot for the binned ratio
sns.countplot(x='maternal_to_adolescent_binned
    ', data=df, palette = "magma")
plt.grid(True, axis='y', color='gray',
    linestyle='--', linewidth=0.5)

plt.title('Count_Plot:_Maternal_Mortality_to_
    Adolescent_Birth_Rate_Ratio_(Binned)')
plt.xlabel('Maternal_to_Adolescent_Ratio')
plt.ylabel('Count')
plt.show()
```

In Fig. 4 the Scatter Plot shows a positive correlation between female and male secondary education, with countries with higher levels of human development tending to have higher levels of secondary education for both genders. In less developed countries, education percentages are lower and gender gaps in access to education are more evident. The following code is how Fig. 4 was obtained:

```
    # Bivariate analysis
# 4. Scatter plot between F_secondary_educ and
    M_secondary_educ
sns.scatterplot(x='F_secondary_educ', y='
    M_secondary_educ',  data=df, hue='HUMAN_
    DEVELOPMENT', palette = "magma")
plt.grid(True, color='gray', linestyle='--',
    linewidth=0.5)
plt.title('Scatter_plot:_Female_vs_Male_
    Secondary_Education_(%)')
plt.show()  # Show plot
```

In Fig. 5, the correlation matrix highlights the relationships between various socioeconomic and health variables. There are strong negative correlations between HDI rank and both female (-0.87) and male (-0.85) secondary education, indicating that a higher HDI is associated with higher levels of education in both sexes. There is a positive correlation between the adolescent birth rate and maternal mortality (0.77), suggesting that a higher adolescent birth rate corresponds to higher maternal mortality. In addition, there is a weak positive correlation between the percentage of women in parliament and female labor force participation (0.25), but limited correlation with other variables. This graph was printed using the following code:

```
    # 5. Correlation matrix heatmap
corr_matrix = df.select_dtypes(include=['
    number']).corr()
sns.heatmap(corr_matrix, annot=True, cmap = '
    magma')
plt.title('Correlation␣Matrix')
plt.show()  # Show plot
```

A. Cleaned data

The final result, after all the process of cleaning and preprocessing, a new CSV file containing all changes and new information is created and stored in Google Drive. First, `df.to_csv()` is the function that stores the Data Frame into a CSV file. Then, is written the address where the file will be stored. And the final parameter is for pandas not to include the column of index. The cleaned dataset is shown in Fig. 7 and be compare with Fig. 6, the uncleaned dataset.

```
# Save the preprocessed data
df.to_csv("/content/drive/MyDrive/datasets/
    gender_inequality_index_clean.csv", index=
    False)
```

V. Justification Of The Method Used For Preprocessing

The methods used for preprocessing the dataset were chosen based on the characteristics of the data and the objectives of the analysis:

A. Removing unnecessary columns

The column "Unnamed: 12" was removed because it contained no meaningful data. Keeping unnecessary columns can confuse and affect the accuracy of any analysis.

```
df.columns
```

```
Index(['HDI rank', 'Country', 'HUMAN DEVELOPMENT', 'GII VALUE', 'GII RANK',
       'Maternal_mortality', 'Adolescent_birth_rate',
       'Seats_parliamentt(% held by women)', 'F_secondary_educ',
       'M_secondary_educ', 'F_Labour_force', 'M_Labour_force', 'Unnamed: 12'],
      dtype='object')
```

```
df.drop("Unnamed:␣12", axis=1, inplace = True)
```

B. Replacing placeholder values

The placeholder values ('..') were replaced with `NaN`. This step is crucial because placeholder values can mislead analysis if treated as valid data. By converting them to `NaN`, we can handle missing values appropriately.

```
    # Basic data cleaning (replace '..' with
        NaN)
df.replace('..', np.nan, inplace=True)
```

C. Elimination of duplicate data

Keeping duplicate data can meaningfully affect the final results, since they will not be accurate, that is why eliminating them was a crucial step for preprocessing this dataset.

```
df = df.drop_duplicates(subset=['HDI␣rank'])
```

D. Converting data types

Columns like `Maternal_mortality` and `Adolescent_birth_rate`, among others, were converted to numeric types. This conversion is necessary to perform mathematical operations and analyses. Incorrect data types can lead to errors in calculations and analysis.

```
    # Convert appropriate columns to numeric
        types (error coercion)
df['Maternal_mortality'] = pd.to_numeric(df['
    Maternal_mortality'], errors='coerce')
df['Adolescent_birth_rate'] = pd.to_numeric(df
    ['Adolescent_birth_rate'], errors='coerce'
    )
df['Seats_parliamentt(%␣held␣by␣women)'] = pd.
    to_numeric(df['Seats_parliamentt(%␣held␣by
    ␣women)'], errors='coerce')
df['F_secondary_educ'] = pd.to_numeric(df['
    F_secondary_educ'], errors='coerce')
df['M_secondary_educ'] = pd.to_numeric(df['
    M_secondary_educ'], errors='coerce')
df['F_Labour_force'] = pd.to_numeric(df['
    F_Labour_force'], errors='coerce')
df['M_Labour_force'] = pd.to_numeric(df['
    M_Labour_force'], errors='coerce')
```

E. Handling missing values

Rows with missing values were dropped using `dropna()`. This approach ensures that the analysis is based only on complete cases, which improves the reliability of the results. While this method may result in some loss of data, it is often preferable when dealing with a dataset where missing values are not systematic.

```
    # Drop rows with missing values
df.dropna(inplace=True)
```

F. Creating New Features

The new feature `maternal_to_adolescent` was created to provide insight into the relationship between maternal mortality and adolescent birth rates. Feature engineering is a common practice in data preprocessing, as it can reveal important patterns and improve the predictive power of models.

```
    # Create a new feature:
        maternal_to_adolescent ratio
df['maternal_to_adolescent'] = df['
    Maternal_mortality'] / df['
    Adolescent_birth_rate']
```

G. Binning data

The binned ratios of maternal to adolescent mortality help to simplify the analysis and visualization of the data. Binning can enhance interpretability by grouping continuous data into discrete categories, making it easier to identify trends.

```
    # 2. Bin the ratios into categories
bins = [0, 1, 5, 10, 20, 50, np.inf]
labels = ['0-1', '1-5', '5-10', '10-20', '
    20-50', '>50']
df['maternal_to_adolescent_binned'] = pd.cut(
    df['maternal_to_adolescent'], bins=bins,
    labels=labels)
```

VI. Conclusion

The data processing for the Gender Inequality Index was needed to prepare the data for analysis. The goal was to clean the data by removing unnecessary columns, converting some data into numeric types, and handling missing values. Missing data were replaced with `NaN`, certain columns were converted to numeric characters, and incomplete rows were removed. There was also created a new column with the ratio between maternal mortality and adolescent birth rate and divided it into categories to make it easier to analyze.

These steps were important to make the data clear and ready for analysis, allowing us to get reliable results about gender inequality. Now, the dataset is ready for deeper and more useful analysis.

VII. File Of Raw And Preprocessed Data

The raw data can be found at the following link: https://drive.google.com/file/d/1CiqAWk4ipx-cN7DToHQfgd7Fok3_R0We/view?usp=sharing

The preprocessed data can be found at: https://drive.google.com/file/d/1_zOdTwhoaZr71-y8biTpvpLAMI22aLUi/view?usp=sharing

Appendix

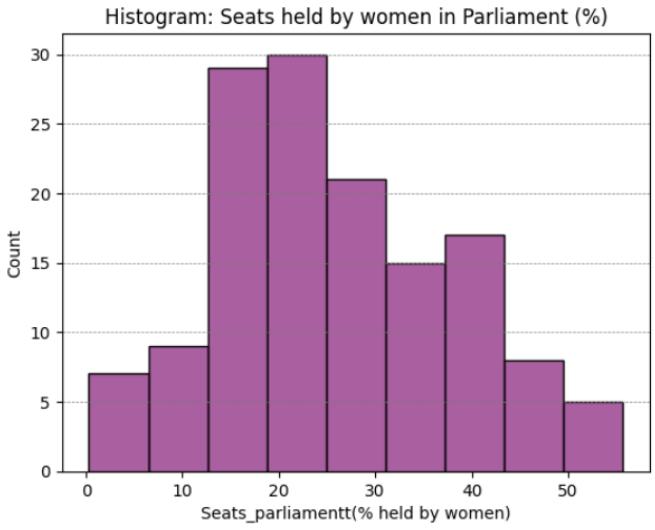

Fig. 1. Histogram: Seats held by women in parliament

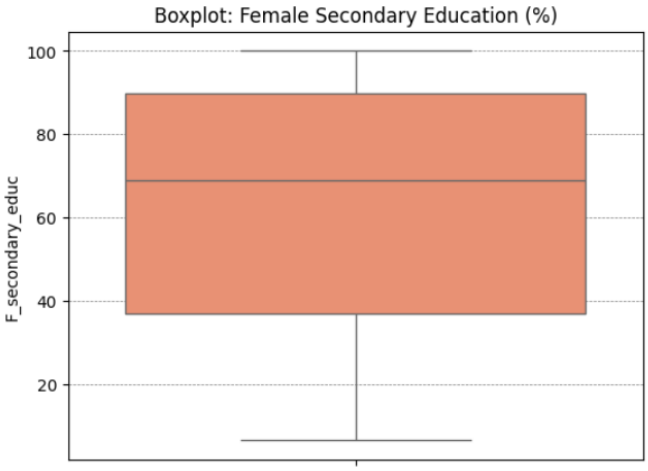

Fig. 2. Boxplot: Female Secondary Education

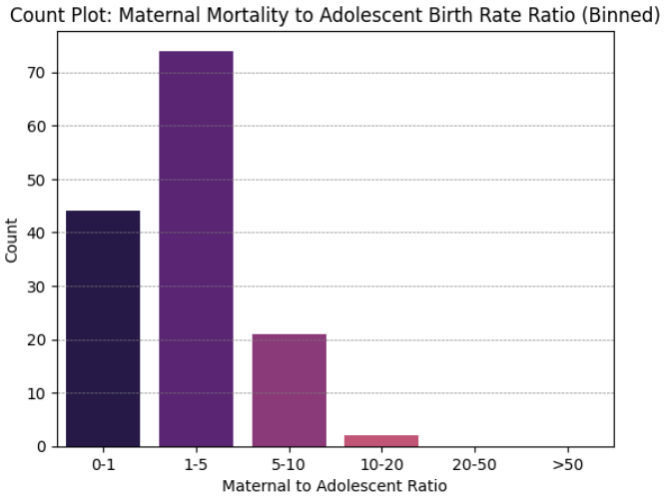

Fig. 3. Count plot: Maternal Mortality to adolescent birth rate ratio

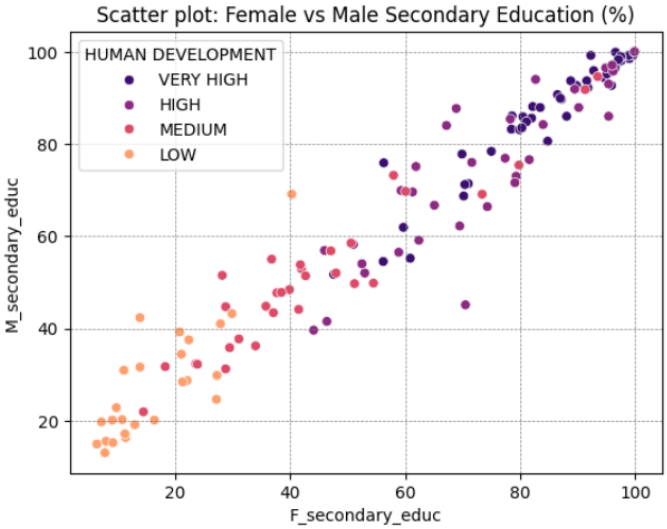

Fig. 4. Scatter plot: Female vs male secondary education

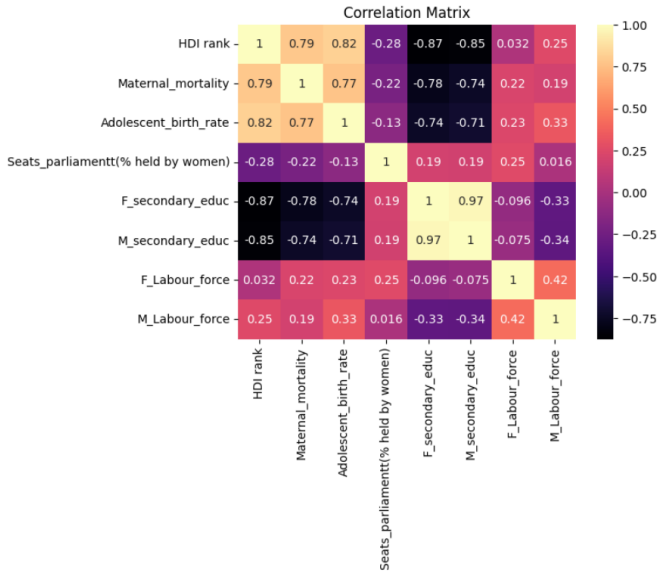

Fig. 5. Correlation matrix

HDI rank	Country	HUMAN DEVELOPM	GII VALUE	GII RANK	Maternal_mortality	Adolescent_birth_rat	Seats_parliamentt(%	F_secondary_educ	M_secondary_educ	F_Labour_force	M_Labour_force
1	Switzerland	VERY HIGH	0.018	3	5	2.2	39.8	96.9	97.5	61.7	72.7
2	Norway	VERY HIGH	0.016	2	2	2.3	45	99.1	99.3	60.3	72
3	Iceland	VERY HIGH	0.043	8	4	5.4	47.6	99.8	99.7	61.7	70.5
4	Hong Kong, China (S	VERY HIGH	..	..	..	1.6	..	77.1	83.4	53.5	65.8
5	Australia	VERY HIGH	0.073	19	6	8.1	37.9	94.6	94.4	61.1	70.5
6	Denmark	VERY HIGH	0.013	1	4	1.9	39.7	95.1	95.2	57.7	66.7
7	Sweden	VERY HIGH	0.023	4	4	3.3	47	91.8	92.2	61.7	68
8	Ireland	VERY HIGH	0.074	21	5	5.9	27.3	88.1	86	56.5	68.6
9	Germany	VERY HIGH	0.073	19	7	7.5	34.8	96.1	96.5	56.8	66
10	Netherlands	VERY HIGH	0.025	5	5	2.8	39.1	89.8	92.7	62.4	71.3
11	Finland	VERY HIGH	0.033	6	3	4.2	46	99	98.5	56.5	64
12	Singapore	VERY HIGH	0.04	7	8	2.6	29.8	80.5	85.9	59.4	76.8
13	Belgium	VERY HIGH	0.048	10	5	5.3	42.9	87.2	89.7	49.8	58.8
13	New Zealand	VERY HIGH	0.088	25	9	12.6	49.2	82	81.8	65.1	75.3
15	Canada	VERY HIGH	0.069	17	10	7	34.4	100	100	60.8	69.7
16	Liechtenstein	VERY HIGH	..	..	..	3	28	..	..	..	..
17	Luxembourg	VERY HIGH	0.044	9	5	4.3	35	100	100	58.5	65.5
18	United Kingdom	VERY HIGH	0.098	27	7	10.5	31.1	99.8	99.8	58	67.1
19	Japan	VERY HIGH	0.083	22	5	2.9	14.2	95.9	92.7	53.3	71
19	Korea (Republic of)	VERY HIGH	0.067	15	11	2.2	19	83.1	93.1	53.4	72.4
21	United States	VERY HIGH	0.179	44	19	16	27	96.5	96.4	55.2	66.4
22	Israel	VERY HIGH	0.083	22	3	7.6	28.3	91.6	93.7	58.5	66.1
23	Malta	VERY HIGH	0.167	42	6	11.5	13.4	82.2	88.1	53.1	71.4
23	Slovenia	VERY HIGH	0.071	18	7	4.5	21.5	97.6	98.7	53.8	62.2
25	Austria	VERY HIGH	0.053	12	5	5.5	39.3	100	100	55.5	66.3

Fig. 6. Uncleaned dataset. First 25 rows without cleaning

HDI rank	Country	HUMAN DEVELOPM	GII VALUE	GII RANK	Maternal_mortality	Adolescent_birth_rat	Seats_parliamentt(%	F_secondary_educ	M_secondary_educ	F_Labour_force	M_Labour_force	maternal_to_adoles	maternal_to_adoles
1	Switzerland	VERY HIGH	0.018	3	5	2.2	39.8	96.9	97.5	61.7	72.7	2.272727273	1-5
2	Norway	VERY HIGH	0.016	2	2	2.3	45	99.1	99.3	60.3	72	0.8695652174	0-1
3	Iceland	VERY HIGH	0.043	8	4	5.4	47.6	99.8	99.7	61.7	70.5	0.7407407407	0-1
5	Australia	VERY HIGH	0.073	19	6	8.1	37.9	94.6	94.4	61.1	70.5	0.7407407407	0-1
6	Denmark	VERY HIGH	0.013	1	4	1.9	39.7	95.1	95.2	57.7	66.7	2.105263158	1-5
7	Sweden	VERY HIGH	0.023	4	4	3.3	47	91.8	92.2	61.7	68	1.212121212	1-5
8	Ireland	VERY HIGH	0.074	21	5	5.9	27.3	88.1	86	56.5	68.6	0.8474576271	0-1
9	Germany	VERY HIGH	0.073	19	7	7.5	34.8	96.1	96.5	56.8	66	0.9333333333	0-1
10	Netherlands	VERY HIGH	0.025	5	5	2.8	39.1	89.8	92.7	62.4	71.3	1.785714286	1-5
11	Finland	VERY HIGH	0.033	6	3	4.2	46	99	98.5	56.5	64	0.7142857143	0-1
12	Singapore	VERY HIGH	0.04	7	8	2.6	29.8	80.5	85.9	59.4	76.8	3.076923077	1-5
13	Belgium	VERY HIGH	0.048	10	5	5.3	42.9	87.2	89.7	49.8	58.8	0.9433962264	0-1
15	Canada	VERY HIGH	0.069	17	10	7	34.4	100	100	60.8	69.7	1.428571429	1-5
17	Luxembourg	VERY HIGH	0.044	9	5	4.3	35	100	100	58.5	65.5	1.162790698	1-5
18	United Kingdom	VERY HIGH	0.098	27	7	10.5	31.1	99.8	99.8	58	67.1	0.6666666667	0-1
19	Japan	VERY HIGH	0.083	22	5	2.9	14.2	95.9	92.7	53.3	71	1.724137931	1-5
21	United States	VERY HIGH	0.179	44	19	16	27	96.5	96.4	55.2	66.4	1.1875	1-5
22	Israel	VERY HIGH	0.083	22	3	7.6	28.3	91.6	93.7	58.5	66.1	0.3947368421	0-1
23	Malta	VERY HIGH	0.167	42	6	11.5	13.4	82.2	88.1	53.1	71.4	0.5217391304	0-1
25	Austria	VERY HIGH	0.053	12	5	5.5	39.3	100	100	55.5	66.3	0.9090909091	0-1

Fig. 7. Cleaned dataset. First 25 rows with cleaning